Handwritten Digit Recognition Based on Pooling SVM-Classifiers Using Orientation and Concavity Based Features

Jose M. Saavedra

Orand S.A.
Estado 360, Santiago, Chile
jose.saavedra@orand.cl

Abstract. In order to increase the performance in the handwritten digit recognition field, researchers commonly combine a variety of features to represent a pattern. This approach has showed to be very effective in practice. The classical approach to combine features is by concatenating the underlying feature vectors. A drawback of this approach is that it could generate high-dimensional descriptors, which increases the complexity of the training process. Instead, we propose to use a pooling based classifier, that allow us to get not only a faster training process but also outperforming results. For evaluation, we used two state-of-the-art handwritten digit datasets: CVL and MNIST. In addition, we show that a simple rectangular spatial division, that characterize our descriptors, yields competitive results and a smaller computation cost with respect to other more complex zoning techniques.

1 Introduction

Digit recognition is still an active research area in the document analysis and recognition community due to the high variability produced by factors like noise, image quality and handwriting styles. This variability becomes stronger when we deal with real applications as in the case of bank check processing. Recently, a new competition on handwritten digit recognition was performed, in which a new free handwritten digit dataset "CVL dataset" was released [6]. Different from traditional digit datasets like MNIST [13] or USPS [9] the CVL dataset presents more variability which makes the classification a harder problem.

A fast and effective method for handwritten digit recognition will also produce a positive impact in the performance of other handwriting recognition problems such as bank check processing, automatic form processing, or postal mail sorting [14]. It is worthy to note that industrial applications require recognition methods to show not only high accuracy but also high speed in the classification process in order to be applied in real time environments.

Feature extraction is a critical step in the classification process. According to the literature, feature extractors based on gradient orientations [14,15] are the most used in the community because of their demonstrated high performance.

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 658–665, 2014.

In addition, to improve the performance of the classification, complementary features, like concavities, have been explored [1,18,14].

A common approach to increase the performance in the recognition task is by dividing the image into local regions. This allow us to exploit the spatial distribution of strokes. This technique knows as spatial division or image zoning has shown high performance in the computer vision field [12]. In the case of handwriting recognition, this technique consists in dividing a pattern image into zones or regions and extracting relevant characteristics from each zone. Many zoning methods have been proposed based on different partitioning strategies [10]. In this regard, Impedovo et al. [10] recently proposed a zoning strategy using Voronoi tessellation. They showed that a Voronoi based division is better that a classical rectangular division. However, the task of computing the seed points for the Voronoi division entails an additional computation cost.

Another approach to increase the performance of the classification task is by combining multiple features that describe different characteristics of a pattern. A common approach to this end is by concatenating the feature vectors produced by different feature extractor techniques [18,14]. Although, the concatenation may significantly boost the classification performance, it could yield very large descriptors which could increase the complexity of the training process.

Therefore, the contribution of this paper is many-fold. We show that combining multiple classifiers, that have been generated using different features, by a pooling approach is a better option that the classical concatenation process. This allow us to get not only a much faster but also an outperforming method for digit classification. In addition, our proposal also allows us to speed-up the training process w.r.t. complex methods like *convolutional networks* [13], for instance. Moreover, we present different techniques for the features extractor task. These are based on orientations and concavities, using rectangular spatial division for representing locality. In this vein, we show that a simple spatial division allow us to get competitive results with lower computational cost, in contrast with those based on Voronoi tessellation [10]. We conducted diverse experiments using two standard public databases.

The rest of this document is organized as follows: in Section 2 we describe the approaches to extract features from digit images. Section 3 describes in detail the proposes strategy for classification. Section 4 discusses the experimental evaluation. Finally, Section 5 summarizes the reached conclusions.

2 Feature Description

Orientation is one of the most used features to represent objects in tasks like object recognition, object detection, or object categorization. In fact, state-of-the-art computer vision descriptors such as SIFT [16,11], SURF [2] and HOG [5,7] are based on gradient orientations. Furthermore, considering that digit images are represented only by their strokes, where color and texture are absent, orientation features seems to be a good option to represent those strokes. In fact, this feature seems to be the preferred characteristic in the community [14].

Another feature that has been exploited in the handwriting text recognition community is *the concavity* [18,14,17]. We propose two very simple strategies that are then combined with orientation based strategies. In addition, we use the concavity approach discussed by Heutte et al. [8] and used successfully in the context of numeral string recognition [18].

2.1 Orientation Based Descriptors

We propose two orientation based descriptors. Both of them using a spatial division of the image. The first one is a histogram of orientations computed in a soft manner within the local regions in which the image is divided. The second approach uses the well known HOG descriptor (*Histogram of Oriented Gradients*) that has shown outstanding performance in the computer vision community [5].

1. **Soft Histogram of Gradient Orientations (SHOG)**
   We divide the image into $W \times W$ rectangular regions. For each region, we compute a histogram of gradient orientations with K bins, where each orientation ranges in $[0..2\pi]$. The gradient for each pixel is computed using the Sobel approach. To reduce the negative impact produced by quantizing the gradient angles and by the rectangular division, the histogram of orientations is computed in a soft manner by a tri-linear interpolation process with respect to the gradient angle, the x-axis, and the y-axis.
   The tri-linear interpolation process works as follows:
   - Let p be a pixel in the image, and α_p and mag_p be the corresponding angle and magnitude of the gradient at p, respectively.
   - Compute the two closest bins where α_p falls, and call them A_α, B_α. In addition, compute weights $W_{A_\alpha}, W_{B_\alpha}$ inversely proportional to the distance between α_p and the center of the bins.
   - In the same way, compute the two closest bins where p falls, with respect to x-axis and y-axis. Call them A_x, B_x for the case of x-axis and A_y, B_y for the case of y-axis. In the same manner as in the previous step, compute weights $W_{A_x}, W_{B_x}, W_{A_y}, W_{B_y}$ with respect to the location of p.
   - Update the orientations histograms as follows:

     $$hist_{a,b}(A_\alpha) += W_{A_\alpha} * W_a * W_b * mag_p$$
     $$hist_{a,b}(B_\alpha) += W_{B_\alpha} * W_a * W_b * mag_p$$

     where $a \in \{A_x, B_x\}$, and $b \in \{A_y, B_y\}$, Here, $hist_{a,b}$ represents the histogram for a region defined by a, b.
   - Repeat all the previous steps for each pixel in the digit image.
   - The resulting histograms is normalized to unit. Finally, the resulting SHOG descriptor is produced by the concatenation of all the local histograms.

   The SHOG descriptor requires two parameters to be determined, W, the number of regions in which the image is divided, and K, the number of bins in which the gradient angles are quantized. The K value, also represents the size of each local histogram. Finally the size of the SHOG descriptor is equal to $W \times W \times K$.

2. **Histogram of Oriented Gradients (HOG)**
   This approach is based on the HOG descriptor proposed by Dalal and Triggs [5]. Based on the good experience found in the computer vision community we chose this descriptor to be applied for characterizing digit images. This method works as follows: The image is divided into $C \times C$-size cells. For each cell, a histogram of gradient orientations is computed using K bins. The cell histograms are also computed by a tri-linear interpolation process. After that, the cells are grouped into blocks of $B \times B$ cells. The histogram for each block is then produced by the concatenation of the cell histograms. The block histogram is normalized to unit and the concatenation of all block histograms yields the final HOG descriptor.

2.2 Concavity Based Descriptors

1. **4-Connected Concavity (4CC)**
   We represent each background pixel (white pixel) as a 4-bit concavity code. For coding, we look for foreground pixels (black pixels) in four directions: *north*, *south*, *east*, and *west*.Each direction is associated with a bit position. Thereby, if a foreground pixel is found in a certain direction, the corresponding bit is set to 1, in other case, is set to 0. We, then, form frequency histograms of concavity codes. This leads to a 16-size histogram. To take into account spatial information of the digit image, we also divide the image into $W \times W$ regions. For each region we compute a 16-bin concavity histogram. Therefore, the size of our descriptor is $W \times W \times 16$.
2. **8-Connected Concavity (8CC)**
   This works similar as the previous descriptor. However, in this case we use the four diagonal directions: *north-east*, *north-west*, *south-east*, and *south-west*. The diagonals correspond to the 8-connected corner neighbors. In the same way as in the 4CC descriptor, we use spatial division producing a $W \times W \times 16$-size descriptor.
3. **13-bin Concavity (13C)**
   This is a descriptor described by Heutte et al. [8] and used for the numeral string recognition problem by Oliveira et al. [18]. In this case a 13-bin histogram is build as follows: For each white pixel they search for foreground pixels following the 4-Freeman directions. The number of foreground pixels reached and the direction in which black pixels are not reached are stored. This information will be useful for the coding process. In the case that black pixels are reached in all directions (closed stroke is found),they check four auxiliary directions to confirm if the white pixel is really inside of a closed curve. With the codes provided by all the white pixels a 13-bin histogram is built. In addition, to take advantage of the spatial division, the 13-bin concavity is applied in local regions. In this case, a 3×2 grid produce the local regions. A detailed description of this concavity strategy can be found in [18].

3 Classification

Our classification is based on the Support Vector Machine [4]. To increase the classification effectiveness we exploit a combined feature approach. Instead of concatenating all vector descriptors and then use only one classifier, we use many classifiers, each one depending on a specific feature set. This has the advantage of a fast training process because the feature dimension is kept slow, in contrast with a concatenation based method that may produce large descriptors.

In our case, given an input digit image, a feature extractor is applied to characterize the input. Then a classifier, trained in the corresponding feature space, is used. It will produce a classification probability for each of the ten possible classes. As we have many feature types, we also have many classifiers. In order to combine the output of the classifiers we use a pooling process. The pooling process receives the score for each class provided from different classifiers and it aggregate all the scores to estimate just one. We investigate different strategies for pooling. Some the most used are SUM, Average, Borda Count, Voting, MAX. However the SUM pooling was the one that showed a better performance.

4 Experimental Evaluation

4.1 Datasets

We use two state-of-the-art datasets: MNIST, composed of 60000 images for training and 10000 images for testing, and CVL-digits [6] with 7000 images for training, 7000 images for validation and 21780 images for testing.

Here, we demonstrate that a pooling based classification is a better option that a concatenation process. We demonstrate that the pooling proposal achieved a low error rate in the CVL database and competitive results in the MNIST database. In addition, we demonstrate that the training process time reduces significantly when we use pooling classifiers. Indeed, during our experiments in the CVL dataset we observe that when combine SHOG, 4CC, 8CC and HOG, the training time required for the concatenation approach was 155 seconds, while the time required by the pooling classifier was only 57 seconds. This represents a reduction of 64% in the time required for training.

For feature extraction we use SHOG, HOG, 4CC, 8CC, 13C. The results achieved using classifiers that were trained with only one feature set for the CVL database are shown in Table 1, and for the MNIST database, are shown in Table 2. Each descriptor (feature extractor technique) is shown together with their best parameters, the corresponding dimension, and the achieved error rate. In both databases, the best performance was achieved by the HOG descriptor, with an error rate (ER) equal to 4.05 in the case of the CVL database and 0.93 in the case of MNIST. According to results, the CVL database seems to be a more challenging dataset. For the classification task we use **libSVM** [3] using a RBF kernel with $C = 32$ and $\gamma = 0.5$. The value of these parameters were fixed by cross validation.

Table 1. Error Rate (ER) using only one feature extractor method on the CVL database

Feature	Best Parameters	Dim	ER (%)
SHOG	W=5 , K=16	400	4.33
HOG	C=8, B=3, K=9	324	4.05
4CC	W=3	144	5.71
8CC	W=3	144	5.35
13C	-	78	8.00

Table 2. Error Rate (ER) using only one feature extractor method on the MNIST database

Feature	Best Parameters	Dim	ER (%)
SHOG	W=6, K=16	576	1.24
HOG	C=8, B=4, K=18	1152	0.93
4CC	W=2	64	1.84
8CC	W=2	64	1.86
13C	-	78	5.7

As we can see in Tables 1 and 2, the performance of classifiers using only one feature set is low, except for the HOG descriptor in the MNIST database, that presents an outstanding performance by itself (error rate = 0.93%). The performance of single features can be increased using a combination o feature extractor techniques. Following, we evaluate the performance of the concatenation and pooling strategies.

In Table 3 we show the error rate achieved by the proposed pooling classifier for a diverse combination of classifiers in the CVL dataset. We can see that using the five feature extractor techniques (SHOG, HOG, 4CC, 8CC, and 13C) we obtain the best performance (error rate = 3.04%). In addition, in the same table we could see the goodness of our proposal over the concatenation based approach. It is worthy to mention that the CVL database contains separately a training set and a validation set. The achieved results shown in Table 3 were obtained using only the training set to train the classifiers and the validation set to fix parameters. However, if we use both, the validation and the training set, to train the classifiers plus a cross validation process, the performance of our proposal on the testing set increases, obtaining an error rate equal to 2.14%.

In Table 4 the results of combining features using the MNIST database is shown. The error rate of the pooling based classifier for the first two combinations is a little lower that the results of the concatenation approach, but still competitive. However, for the best option, that is combining four feature extractors SHOG, 4CC, 8CC, and HOG, the pooling approach outperforms the concatenation strategy.

About the pooling strategy, we use SUM pooling because it outperforms other techniques like MAX, or Borda Count. In addition, in Table 5 we compare the results achieved by the SUM pooling against those achieved by the MAX pooling in the CVL database.

Finally, our results also show that our feature extractor techniques, that are based on rectangular spatial division, are competitive with the result of using Voronoi based zoning [10], whose best result in the MNIST dataset is 0.77%. Our best result in that dataset is 0.81% but without requiring an additional cost for zoning.

Table 3. Error Rate (ER) of digit classification on the CVL database by concatenating features (2nd column) and by SUM pooling (3rd column)

CVL-Database Features	Conc. ER (%)	SUM ER (%)
4CC+8CC	5.54	4.85
SHOG+4CC+ 8CC	3.82	3.57
SHOG+4CC+ 8CC+HOG	3.82	3.17
SHOG+4CC+ 8CC+HOG+13CC	3.77	3.04

Table 4. Error Rate (ER) of digit classification on the MNIST database by concatenating features (2nd column) and by pooling classifiers (3rd column)

MNIST-Database Features	Conc. ER (%)	SUM ER (%)
4CC+8CC	1.32	1.55
SHOG+4CC+ 8CC	1.04	1.15
SHOG+4CC+ 8CC+HOG	1.06	0.81

Table 5. Error Rate using MAX pooling and SUM pooling on the CVL database

CVL-Database Features	MAX Pooling (%) ER (%)	SUM Pooling ER (%)
4CC+8CC	6.58	4.85
SHOG+4CC+8CC	3.82	3.57
SHOG+4CC+8CC+HOG	3.50	3.17
SHOG+4CC+8CC+HOG+13CC	3.51	3.04

5 Conclusions

We have presented an alternative approach for combining features in the context of handwritten digit recognition. The proposal is based on using multiple classifiers, each one generated from different features, and aggregating their responses. This scheme may work easily in parallel environments. Our results show that the pooling approach is the better option for the CVL-dataset and is very competitive in the case of MNIST database. In addition, the training process of the proposal is faster with respect to the concatenation approach.

Our ongoing work is focused on evaluating other feature extractors to improve the achieved results and on to apply our approach for handwritten character recognition.

References

1. Azeem, S.A., El Meseery, M.: Arabic handwriting recognition using concavity features and classifier fusion. In: Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops, ICMLA 2011, vol. 01, pp. 200–203 (2011)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)

3. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2(3), 27:1–27:27 (2011)
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20(3), 273–297 (1995)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE Computer Society (2005)
6. Diem, M., Fiel, S., Garz, A., Keglevic, M., Kleber, F., Sablatnig, R.: Icdar 2013 competition on handwritten digit recognition (hdrc 2013). In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1422–1427 (2013)
7. Felzenszwalb, P., David, M., Deva, R.: A discriminatively trained, multiscale, deformable part model. In: International Conference on Computer Vision and Pattern Recognition (2008)
8. Heutte, L., Moreau, J., Plessis, B., Plagnaud, J., Lecourtier, Y.: Handwritten numeral recognition based on multiple feature extractors. In: Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 167–170 (1993)
9. Hull, J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(5), 550–554 (1994)
10. Impedovo, S., Mangini, F., Pirlo, G., Barbuzzi, D., Impedovo, D.: Voronoi tessellation for effective and efficient handwritten digit classification. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 435–439 (2013)
11. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 506–513. IEEE Computer Society (2004)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178 (2006)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
14. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition 36(10), 2271–2285 (2003)
15. Liu, H., Ding, X.: Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition, vol. 1, pp. 19–23 (2005)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
17. Karic, M., Martinovic, G.: Improving offline handwritten digit recognition using concavity-based features. International Journal of Computers, Communications & Control 8(2), 220–234 (2013)
18. Oliveira, L., Sabourin, R., Bortolozzi, F., Suen, C.: Automatic recognition of handwritten numerical strings: a recognition and verification strategy. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(11), 1438–1454 (2002)